CHOUETTE

Maintenance, accompagnement et recette de logiciels pour les échanges de données multimodales

application Chouette
-
manuel d'installation (V2.4)

Auteurs :	Michel ETIENNE, Luc DONNET, Marc Florisson (CityWay)
Relecteurs	Patrick GENDRE (CEREMA), Jean SENG (AFIMB)

Résumé :

CHOUETTE est un logiciel libre développé à l'initiative du ministère français chargé des transports (et du développement durable) dans le but de faciliter l'échange de données d'offre (théorique) de transport collectif (TC), en s'appuyant pour cela sur la norme NFP 99506, dite Neptune, qui spécifie un profil d'échange XML.

Les utilisateurs visés sont les collectivités locales Autorités Organisatrices de Transport (AOT), les exploitants des réseaux TC, ainsi que leurs prestataires (bureaux d'étude ou société de services) et d'autres acteurs tels que services de l'état, éditeurs de logiciels, opérateurs de services d'information, chercheurs...

L'application CHOUETTE est disponible sous la forme d'une application WEB destinée à être déployée sur un serveur WEB, et d'une application en ligne de commande.

Ce manuel d'installation liste les logiciels à installer, puis les instructions pour créer la base de données, installer et configurer l'application web.

L'utilisation depuis une console Ruby est également documentée.

Agence française pour l'information multimodale et la billettique

Informations sur ce document :

<table>
<tr><td colspan="4">Organisme commanditaire : AFIMB</td></tr>
<tr><td colspan="4">Titre : Manuel d'installation de l'application CHOUETTE</td></tr>
<tr><td colspan="4">Sous-titre : Procédure à suivre pour installer et configurer l'application CHOUETTE.</td></tr>
<tr><td colspan="2">Organismes auteurs
CITYWAY
CEREMA DT Med.

Maitre d'ouvrage
AFIMB</td><td>Rédacteurs
Marc FLORISSON
Zakaria BOUZIANE
Michel ETIENNE
Luc DONNET</td><td>Participants
Patrick GENDRE
Jean SENG</td></tr>
<tr><td colspan="2">Mots clés :
profil d'échange Neptune, information multi-modale, application Chouette, manuel d'installation, postgreSQL, JAVA , Ruby</td><td colspan="2">Diffusion :

publique (licence Creative Commons CC-by-nd)</td></tr>
<tr><td>Nombre de pages :
15 pages</td><td>Date
Mai 2014</td><td>Confidentialité :
Non</td><td>Bibliographie :
Oui</td></tr>
</table>

Historique des versions / révisions :

Version	Date d'application	Description des changements	Auteur
2.0.0 2.0.2	19/09/2012 15/01/13	Installation environnement Jruby refonte du plan et mise à jour	Michel Etienne Patrick Gendre
2.0.3	04/2013	Mise à jour version 2.0.3	Michel Etienne
2.0.4	07/05/13	Ajout du chapitre 3.7	Marc Florisson
2.0.5	28/05/13	Ajout du §4.4.2 pour la configuration de l'accès au Géoportail ; Ajout du document Modele-classes.pdf en ressource externe ; Référencement de ce document au chapitre 3.7	Marc Florisson
2.0.6	31/05/13	complément mise à jour de version	Patrick Gendre
2.0.7	20/06/13	Corrections diverses suites aux remarques	Michel Etienne
2.1.0	15/10/13	mise à jour	Patrick Gendre
2.2.0	28/02/2014	Suppression de jruby et tomcat	Michel Etienne
2.3	Mars 2014	Migration technique Java	Michel Etienne

Version	Date d'application	Description des changements	Auteur
2.4	Mai 2015	Train de maintenance n°1 : correction d'anomalies	Michel Etienne

Table des matières

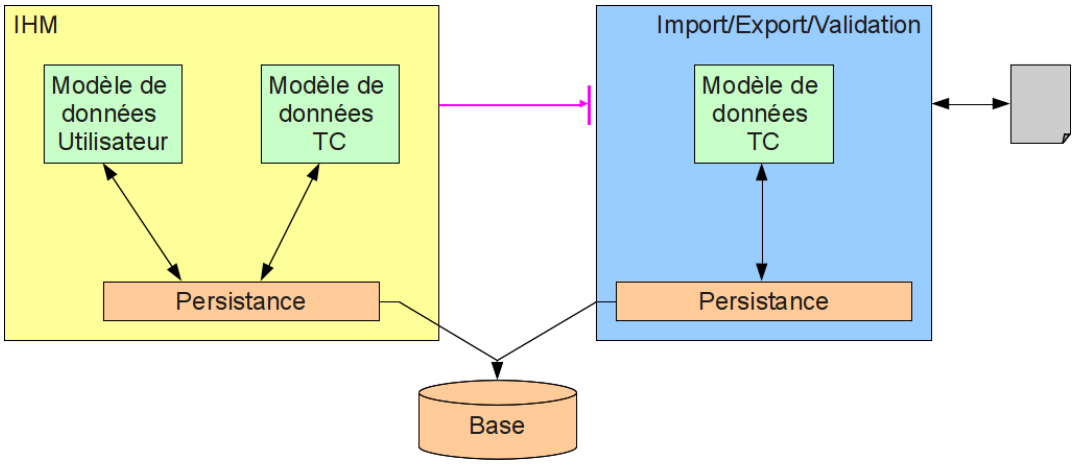
1Vue d'ensemble....4
2Prérequis....5
2.1.Logiciels....5
2.2.Dépendances externes....6
3Migration depuis une version précédente....7
3.1.Version V1.x vers une version 2.x....7
3.2.Version V2.x vers V2.y....7
4Installation....7
4.1.Récupération des sources....7
4.2.Installation du programme d'import/export/validation....8
4.3.Installation du serveur WEB....8
5Configuration....9
6Lancement de l'application....10
7Annexe : Utilisation de la console Rails....11
7.1.Présentation du modèle de données en persistance....11
7.2.Exemples....12

1 VUE D'ENSEMBLE

Ce document décrit comment installer l'application Chouette, mais également comment l'utiliser depuis un interpréteur Ruby pour manipuler les données métier.

Java et Ruby étant portables, Chouette est compatible avec de multiples plates-formes : Mac, Linux, Windows, etc. Le présent manuel s'adresse à un utilisateur confirmé ; il décrit en détail la procédure d'installation pour la plate forme Linux, ainsi que pour Windows.

Chouette est une application web organisée selon l'architecture suivante :



L'application Chouette comprend une application web Ruby/Rails pour gérer l'IHM (fonctions de gestion des données) et, pour gérer les fonctions d'import/export et validation, un programme Java. Toutes deux mettent à jour une base de données PostgreSQL.

L'installation de l'application nécessite 4 étapes, détaillées dans les Chapitres suivants :

-1- d'avoir installé les librairies et outils (paquets) utilisés par le logiciel

-2- d'installer la base de données

-3- de générer (4a) ou récupérer (4b) les applications compilées (war, jar)

-4- de configurer ces applications

Une documentation de base (« quick install ») des commandes linux à lancer pour installer Chouette est disponible en ligne : https://github.com/afimb/chouette2/blob/V2_2/README.md

La version de linux doit être basée sur un noyau version 2.6 minimum et 64 Bits. (C'est le cas par exemple Ubuntu 13.10).

Au niveau mémoire, il est recommandé de disposer d'une machine ayant au minimum 2 Go de RAM. Pour traiter de grands réseaux comme ceux d'Ile de France, il est nécessaire de disposer d'un minimum de 4Go de Ram et de 4 CPUs (mais attention dans ce cas à utiliser une JVM 64 bits).

2 PRÉREQUIS

2.1. Logiciels

Le tableau ci-dessous dresse la liste des logiciels nécessaires avant de procéder à l'installation de Chouette. Ces composants doivent être installés dans votre environnement suivant leurs propres procédures (si nécessaire, reportez-vous aux procédures et/ou outils d'installation de ces composants)

Logiciel	Version minimale	Version conseillée	Intérêt
Plate forme JRE http://www.java.com/fr/download	1.7 (64 bits)	1.7 (64 bits)	exécution en ligne de commande
Ruby https://www.ruby-lang.org/fr/	1.9.3	1.9.3	Interpreteur Ruby pour l'application WEB
Base PostgreSQL http://www.postgresql.org/	9.1	9.2	persistance des données de l'application CHOUETTE
Proj.4 http://trac.osgeo.org/proj/	4.8.0	4.8.0	Fonctions Géographiques : - pour Linux : packets proj-bin et libproj-dev - pour Windows : proj_lib

Librairie Proj4

Sous wndows, une fois l'installation de OSGEO4W effectuée :

- créer la variable d'environnement PROJ_LIB = [répertoire OSGeo4W]\share\proj
- ajouter au Path : [répertoire OSGeo4W]\bin

Si vous avez choisi une VM 64 bits, installez proj4 aussi en version 64 bits (OSGeo4W64).

2.2. Dépendances externes

En complément des logiciels nécessaires au déploiement, si vous souhaitez générer les exécutables à partir des sources, il faudra en outre installer les logiciels suivants pour linux (sous Windows nous n'avons pas documenté la génération des exécutables).

Logiciel	Version minimale	Version conseillée	Intérêt
Plate forme jdk http://www.oracle.com/technetwork/java/javase/downloads/index.html	1.7	1.7	Compilation java
Git pour récupérer les modules sur github http://git-scm.com/	1.7.9	1.7.10	accès au référentiel des sources de Chouette
RVM pour installer Ruby https://www.digitalocean.com/community/articles/how-to-install-ruby-on-rails-on-ubuntu-12-04-lts-precise-pangolin-with-rvm	1.24.7	1.24.7	Le lien indique l'installation de la dernière version stable

Par exemple sous Linux base Debian , installer les packages linux suivants

sudo apt-get install postgresql

sudo apt-get install libpq-dev

sudo apt-get install openjdk-7-jdk

sudo apt-get install git

sudo apt-get install unzip

sudo apt-get install proj-bin

sudo apt-get install libproj-dev

sudo apt-get install make

Installer RVM soit par apt-get install ruby-rvm si la distribution linux en dispose, soit en suivant la procédure fournie dans le lien fourni dans le tableau.

Installer ruby 1.9.3

```
rvm  install ruby-1.9.3-p448
rvm --default use 1.9.3-p448
```

3 MIGRATION DEPUIS UNE VERSION PRÉCÉDENTE

3.1. Version V1.x vers une version 2.x

La version 2 correspond à une réécriture complète de l'IHM dans le framework Ruby on Rails. Le schéma de la base 2.0 est totalement modifié pour s'adapter aux règles de Rails (Active Records); de plus, la gestion des organisations, des utilisateurs, et des espaces de données ne permettent pas de réaliser une procédure de migration de schéma..

Pour récupérer vos données gérées avec une version antérieure à la V2 dans la nouvelle version, il faut donc exporter l'ensemble des réseaux au format Neptune, puis importer les fichiers Neptune dans les espaces de données adéquats depuis Chouette V2.

3.2. Version V2.x vers V2.y

La procédure d'installation décrite ci-après convient aussi bien pour une première installation que pour une mise à jour de version.

La descente de version n'est pas garantie (il peut y avoir des pertes de données)

4 INSTALLATION

L'installation décrite ci-après suppose d'être faite avec un compte utilisateur linux dédié (exclure root) , celui-ci est nommé `myuser:mygroup` dans la suite du document.

Ce compte doit disposer du droit 'sudoer'

4.1. Récupération des sources

Cette section présente la procédure pour générer les exécutables de l'application web et du programme java chouette-command, utilisé par la web pour les imports / exports / validations. Il est rédigé **uniquement pour linux** ; la procédure devrait fonctionner pour Windows compte tenu de la portabilité de Java et Ruby, mais elle n'est pas documentée ici.

Avant de récupérer les sources disponibles sur github, nous vous invitons à vous enregistrer sur le site internet www.chouette.mobi dans la rubrique CHOUETTE / Sources, ce qui nous permet de mieux connaitre nos utilisateurs.

L'archive contenant les sources est accessible sur github aux adresses :

http://github.com/afimb/chouette : code java en ligne de commande (branche V2_3)

http://github.com/afimb/chouette2: code ruby de l'ihm web (branche V2_2)

http://github.com/afimb/ninoxe: code ruby du modèle de donnée Neptune (branche V2_2). Le module (gem) Ninoxe n'apparait pas dans la procédure d'installation car il est inclus dans les dépendances de chouette2.

Attention : les versions stables de Chouette et Chouette2 sont celles qui portent un TAG ; se référer au dernier TAG disponible sur Github. La version à date d'édition de ce document est la V2.4.0 pour Chouette et Chouette2.

4.2. Installation du programme d'import/export/validation

Le programme `chouette_gui_command` doit être installé afin d'exécuter les tâches d'import, export et validation

```
sudo mkdir -p /usr/local/opt/chouette-command/

sudo chown -R myuser:mygroup /usr/local/opt/chouette-command/

cd /usr/local/opt/chouette-command/

wget http://maven.chouette.cityway.fr/fr/certu/chouette/chouette-gui-command/2.4.0/chouette-gui-command-2.4.0.zip

unzip chouette-gui-command-2.4.0.zip

cd chouette-cmd_2.4.0

sudo chmod a+w .
```

4.3. Installation du serveur WEB

1- Récupération des sources sous github

```
cd [working_directory]

wget https://github.com/afimb/chouette2/archive/V2.4.0.zip

unzip V2.4.0.zip
```

2- Télécharger les librairies externes (gems) :

```
cd chouette2

gem install bundler (pour une nouvelle installation)

bundle install
```

3- Préparer la base de données (pour une nouvelle installation)

création du compte d'exploitation : chouette

→ créer l'utilisateur `chouette`, password `chouette`, privilèges `admin`

```
sudo su - postgres

createuser -d -P -U postgres chouette

Enter password for new role: chouette

Enter it again: chouette

Shall the new role be a superuser? (y/n) y

exit
```

4- Initialiser ou mettre à jour la base de données

```
RAILS_ENV=production bundle exec rake db:create apartment:migrate
```

5- Préparer les données statiques

```
RAILS_ENV=production bundle exec rake assets:clean assets:precompile
```

6- Préparer les répertoires d'exploitation : (pour une nouvelle installation)

Les valeurs utilisées sont fixées dans le fichier `config/environments/production.rb` :

- ImportTask.root
- ExportTask.root

Création des répertoires sous le compte `myuser:mygroup` :

```
sudo mkdir -p /var/lib/chouette/imports
sudo mkdir -p /var/lib/chouette/exports
sudo chown myuser:mygroup /var/lib/chouette/imports /var/lib/chouette/exports
sudo chmod a+x /var/lib/chouette/imports /var/lib/chouette/exports
```

5 CONFIGURATION

La configuration est définie dans le fichier `config/environments/production.rb`

Pour les différents paramétrages, il faut se référer aux documents suivants :

- Action Mailer :http://guides.rubyonrails.org/action_mailer_basics.html
- API Géoportail : http://api.ign.fr/accueil
- API Google Analytics : https://www.google.fr/intl/fr/analytics/

Configuration des URLs des vues Action Mailer :

Modifier le paramètre `config.action_mailer.default_url_options`

Configuration de SMTP :

Modifier le paramètre `ActionMailer::Base.smtp_settings`

Configuration de l'adresse E-mail serveur :

Modifier le paramètre `config.mailer_sender`

Configuration de la Clé Géoportail IGN :

Décommenter et fixer l le paramètre `config.geoportail_api_key`

Configuration de Google Analytics :

Modifier le paramètre `GA.tracker`

6 LANCEMENT DE L'APPLICATION

L'application pourra être lancée au démarrage de la machine en installant des scripts dans les répertoires adaptés au système d'exploitation (Base RedHat ou Debian).

Dans la suite, le lancement est expliqué pour un démarrage à la demande.

1- Lancer la gestion des tâches asynchrones (import, export et validation) :

```
RAILS_ENV=production bundle exec rake jobs:work
```

2- Lancer le serveur Web : (WEBrick : serveur par défaut de RubyOnRails)

```
RAILS_ENV=production bundle exec rails server
```

Cette action peut être remplacée par un déploiement sous Phusion Passenger sous Apache ou sous Nginx.

Apache et Nginx gèrent les ressources statiques, auquel cas il faut désactiver cette action dans `config/environments/production.rb` en passant le paramètre `serve_static_assets` à `'false'`

7 ANNEXE : UTILISATION DE LA CONSOLE RAILS

Le framework RubyOnRails fournit un certain nombre de facilités en mode commande: http://guides.-rubyonrails.org/command_line.html#rails-console

Il est intéressant d'en tirer parti pour manipuler de manière souple les données métier gérées par Chouette.

Par défaut, l'environnement (au sens des applications RubyOnRails) de la console est "development". Dans ce cas, la console utilise les paramètres de connexion associés à « `development` » dans le fichier `config/database.yml`

L'environnement peut être précisé en option. Par exemple, la commande ci-dessous permet de se connecter en environnement de « production » :

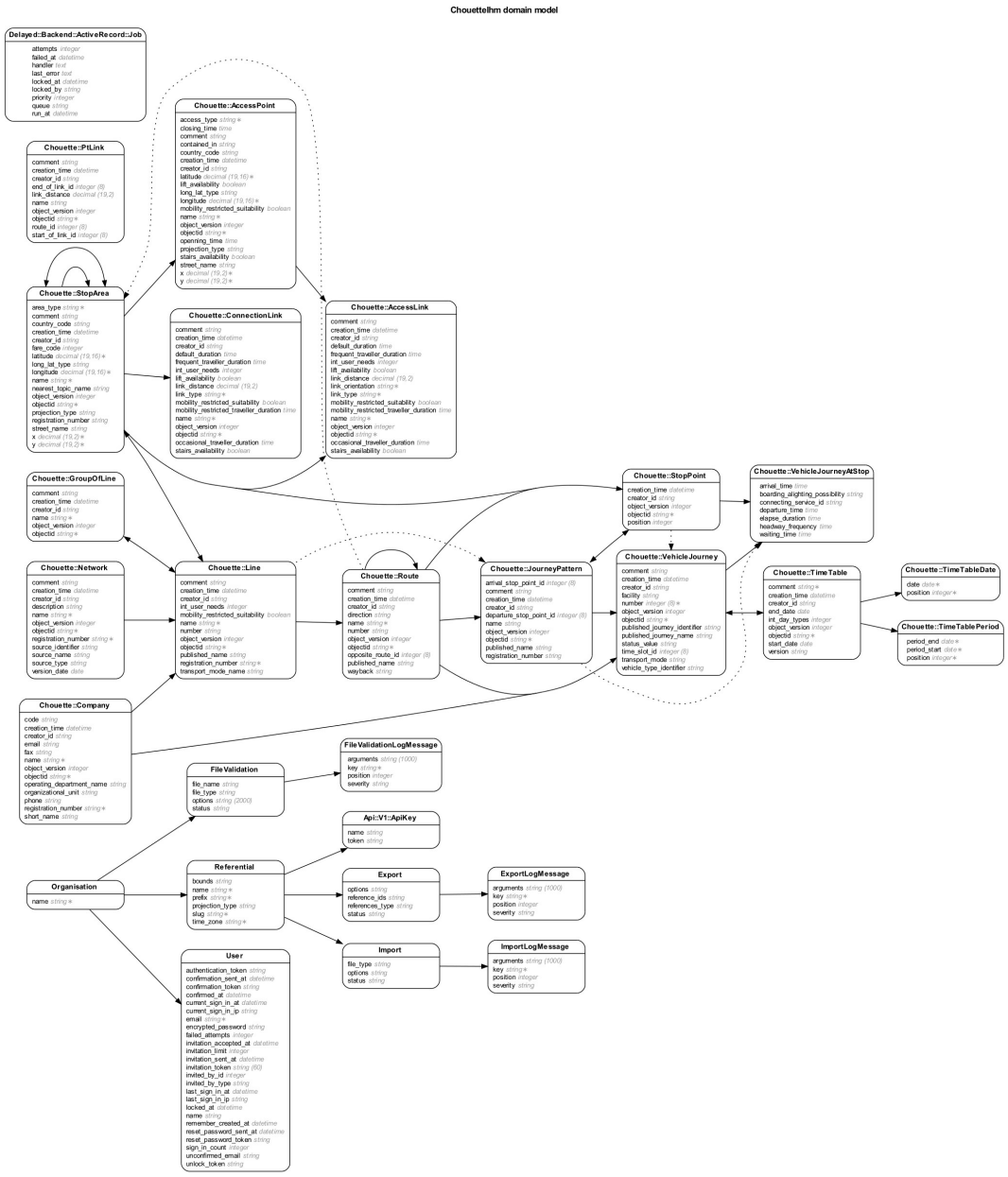
```
bundle exec rails c production
```

Dans la suite de cette annexe, quelques scripts sont détaillés de manière à illustrer quelques cas d'utilisation de la console. Ces scripts ruby peuvent être exécutés dès l'ouverture de la console par copier/coller de chaque ligne de commande.

7.1. Présentation du modèle de données en persistance

Le schéma présenté dans le document « Modele-classes.pdf » introduit les modèles (au sens d'une application Rails) qui sont gérés en persistance.

Le schéma précise les relations entre les modèles (flèches) ainsi que les propriétés de chacun des modèles. Ce schéma est présenté ci-après en petite taille.



7.2. Exemples

Affichage des arrêts commerciaux d'une séquence d'arrêts

```
# Lister les référentiels disponibles

Referential.all.map(&:slug)


# Se placer dans un référentiel de la base

# en supposant que la liste ci-dessus contienne "tatrobus"

Referential.find_by_slug("tatrobus").switch
```

```
# Lister les noms des lignes du réseau
Chouette::Line.all.map(&:name)

# Sélectionner une ligne par son nom
# en supposant que la liste ci-dessus contienne "Ligne 1 Bleue"
line = Chouette::Line.find_by_name( "Ligne 1 Bleue")

# Sélectionner la première séquence d'arrêts
route = line.routes.first

# Lister les noms des arrêts de la séquence
route.stop_areas.map(&:name)

# Lister les noms des arrêts commerciaux relatifs aux arrêts de la séquence
route.stop_areas.map(&:parent).map(&:name)
```

Affichage des horaires de départ d'une séquence d'arrêts

```
# Lister les référentiels disponibles
Referential.all.map(&:slug)

# Se placer dans un référentiel de la base
# en supposant que la liste ci-dessus contienne "tatrobus"
Referential.find_by_slug("tatrobus").switch

# Sélectionner la première séquence d'arrêts
route = Chouette::Line.find_by_name( "Ligne 1 Bleue").routes.first

# Lister les horaires de départ par ordre croissant
route.vehicle_journeys.map {|v|
v.vehicle_journey_at_stops.first.departure_time}.sort
```

Consultation des liens d'accès d'un arrêt

```
# Lister les référentiels disponibles
Referential.all.map(&:slug)
```

```
# Se placer dans un référentiel de la base
# en supposant que la liste ci-dessus contienne "tatrobus"
Referential.find_by_slug("tatrobus").switch

access_link = Chouette::AccessLink.all.first

# afficher l'arrêt
access_link.stop_area

# afficher l'accès
access_link.access_point
```

Vérification d'une date pour un calendrier d'application

```
# Lister les référentiels disponibles
Referential.all.map(&:slug)

# Se placer dans un référentiel de la base
# en supposant que la liste ci-dessus contienne "tatrobus"
Referential.find_by_slug("tatrobus").switch

# Sélectionner un calendrier
tm = Chouette::TimeTable.all.first

# Vérifier si la date du jour est comprise dans le calendrier
tm.include_day?( Date.today)
```